forest management note

Note No. 22 Northern Forest Research Centre Edmonton, Alberta

FORTRAN SUBROUTINES FOR BIOMASS COMPUTATION

Forest biomass is a viable partial alternative to the rapidly depleting fossil fuels in Canada. Biomass estimates are needed to assess the energy potential of this renewable resource, which is abundant throughout the country.

Field techniques vary slightly from place to place, although the basic approach taken is similar in most cases. It is possible to compute biomass data readily and more accurately on a computer than to do lengthy and slow calculations manually. Alemdag (1980) has described the methods for collecting field and laboratory data.

This note describes the FORTRAN subroutines developed at NoFRC to compute the dry weight biomass of a tree and its stem and nonstem components from the field and laboratory data using a PDP 11/60 minicomputer. Guidelines are provided for using the subroutines with subsample data. The programs can be easily adapted to run on any computer that has a FORTRAN compiler.

Because of the size and bulk of entire trees, the field and laboratory procedures used at NoFRC (Singh 1982) are based on a subsampling concept in which disk sections are taken from the merchantable and nonmerchantable stem and nonstem tree components to provide relevant biomass information (Fig 1.)

The field and laborstory data collected for each tree component and the disk subsamples include the following basic items:

Stem and nonstem components
- total fresh weight
- height of the tree at each end of the stem and nonstem components.

Disk samples
- diameter outside bark (dob)
- fresh weight
- dry weight of bark
- dry weight of wood
- height of the tree where the stem disk subsample was taken.

Three main subroutines (RATIOS, STEM, and NSTEM) perform the required computations for determining the dry weights of wood and bark for various components of the tree.

The number of subsamples and the subsampling procedures usually differ among tree components; consequently, the basic computing steps may vary slightly. For stem components, the tree stem samples are taken according to a variable plan: e.g., four subsamples from the merchantable portion ($\geq$ 10 cm dob), three subsamples from the nonmerchantable stem from 10 cm dob up to a minimum specified dob limit (such as $\geq$ 2 cm dob), and one from near the top ($<$ 2 cm dob). The nonstem components can also have a variable sampling plan: e.g., three or more subsamples for each component. The subroutines are written to accommodate any other sampling procedure involving more or fewer subsamples for each main stem and nonstem component, depending on the available storage space in the computer.

In all cases driver programs[1] (FORTRAN statements) have to be written specifically for the three subroutines to perform the needed computations according to the number of subsamples and tree components used in a study.

[1] Sample statements for running these subroutines at NoFRC are listed in Appendix 1.

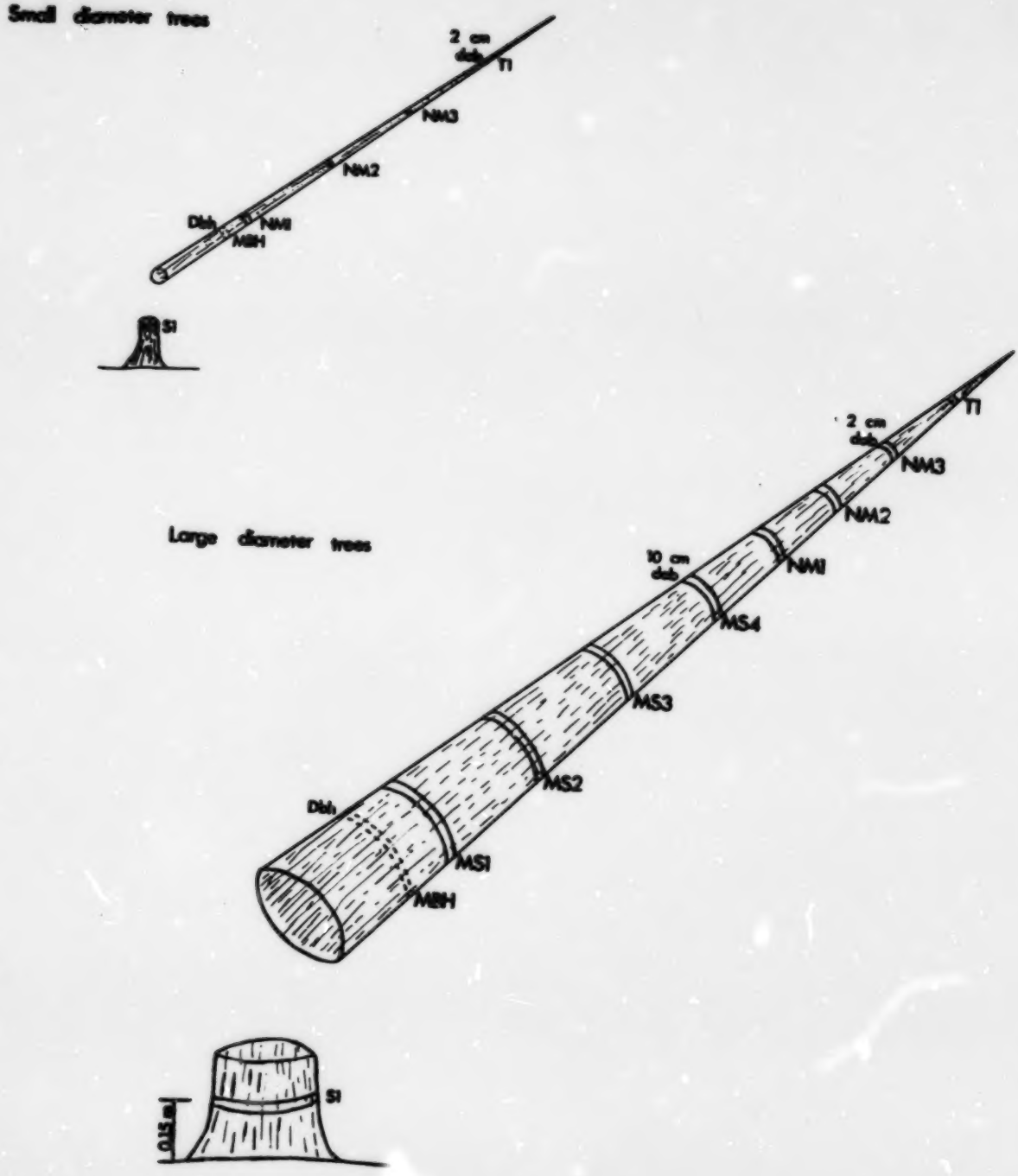

Figure 1. Locations of stem subsamples taken from the m[illegible]antable and nonmerchantable trees.

Guidelines for Program Usage

1. Data preparation, screening, and evaluation are important initial steps in any data analysis. Often there are sporadic items that seem to represent impossible conditions (e.g., dry weight > fresh weight), items that are inadvertently included with the data, or items that are data entry errors. A data base management system such as DATATRIEVE for PDP or SPIRES for IBM 360/370 can be helpful in locating such oddities or errors, which can then be reexamined, corrected, or deleted from the data file. RATIOS subroutine also has a built-in feature for locating such impossible items, which are subsequently dealt with as missing values.

2. The user must provide the FORTRAN statements to run the computer program subroutines listed in this publication. These statements must include information on the number of subsamples taken for each tree component, the total number of trees, and related information according to the subroutine requirements.

3. The results from the subroutine RATIOS should be checked thoroughly for wild points or outliers, which can be helpful in pinpointing data errors. Any such error must be carefully scrutinized and corrected in the data file before running the rest of the program.

4. The running of program segments containing STEM and NSTEM subroutines will give the biomass information on the dry weight of bark and wood for the stem and nonstem components of the tree. This information can then be summarized or used for deriving prediction equations based on the chosen predictor variables.

Comments are included in each subroutine to help the user in understanding the constraints involved and to assist with any desired modifications for individual applications.

Subroutine RATIOS

The ratios R_i and r_i are computed by this subroutine. All computed ratios are to be stored in the computer for later use in STEM and NSTEM subroutines.

The ratios R_i and r_i are determined as follows for each stem and nonstem subsample:

$$R_i = \frac{\text{Dry weight of wood + dry weight of bark}}{\text{Fresh weight of wood and bark}}$$

$$r_i = \frac{\text{Dry weight of bark}}{\text{Dry weight of wood and bark}}$$

If any ratio R_i or r_i is >1, or any value of the entities in the numerator or denominator is zero, the information from that subsample is ignored and treated as a missing value. Such a missing value is then estimated as an average of the two computed adjacent ratios.

A listing of the subroutine is given below.

```
      SUBROUTINE RATIOS(N,MBH,DISKS,RMBH,R,IDB,RDB)
C
C     This subroutine calculates the ratios
C     ( dry-wt / fresh-wt ) and ( dry-bark/ dry-wt )
C     for sample disks
C
C     Variables
C
C     N       - number of disks
C             N(1) - number of disks in stump
C                    1st tree component
C             N(2) - number of disks in merchantable stem
C                    2nd tree component
C             N(3) - number of disks in nonmerchantable stem
C                    3rd tree component
C             N(4) - number of disks in nonmerchantable top
C                    4th tree component
C
C     MBH       - MBH sample disk, this disk is treated
C                 separately because it can be anywhere
C                 in merchantable stem or nonmerchantable
C                 stem
C             MBH(1) - diameter over bark
C             MBH(2) - height
C             MBH(3) - fresh weight
C             MBH(4) - dry weight wood
C             MBH(5) - dry weight bark
C
C     DISKS       - data from sample disks
C             DISKS(i,1,k) - ith sample disk diameter
C                            over bark for kth tree component
C             DISKS(i,2,k) - height
C             DISKS(i,3,k) - fresh weight
C             DISKS(i,4,k) - dry weight wood
C             DISKS(i,5,k) - dry weight bark
C
C     R       - ratios
C             R(i,1,k) - ith sample disk ( dry-wt / fresh-wt )
C                        for kth tree component
C             R(i,2,k) -       ( dry-bark / dry-wt )
C
C     RMBH      - ratios at MBH
C             RMBH(1) - ( dry-wt / fresh-wt )
C             RMBH(2) - ( dry-bark / dry-wt )
C
C     IDB       - debug information
C             IDB(1) = i if R(i,1,k) > 1, 0 otherwise
C             IDB(2) = k if R(i,1,k) > 1, 0 otherwise
C
C     RDB       - debug information
C             RDB = R(i,1,k) if R(i,1,k) > 1, 0 otherwise
C
C           Note assumption of only 1 error per tree
C
      INTEGER N(4)
      REAL MBH(5), DISKS(5,5,4), RMBH(2), R(5,2,4)
      INTEGER IDB(2)
      REAL RDB
C
      REAL DOB, HT, FW, DWW, DWB, BUF(20,2)
      INTEGER I, J, K, II
C
C     Initialize
C
      IDB(1) = 0
      IDB(2) = 0
      RDB = 0.0
      DO 10, K = 1, 4
        DO 10, J = 1, 5
          DO 10, I = N(K), 5
            IF ( I .NE. N(K) ) THEN
              DISKS(I,J,K) = 0.0
              IF ( J .LE. 2 ) THEN
                R(I,J,K) = 0.0
              ENDIF
            ENDIF
10    CONTINUE
C
C     Calculate R values
C
      FW = MBH(3)
      DWW = MBH(4)
      DWB = MBH(5)
      IF ( FW .EQ. 0.0 .OR. DWW .EQ. 0.0
     X     .OR. DWB .EQ. 0.0 ) THEN
        RMBH(1) = 0.0
      ELSE
        RMBH(1) = ( DWW + DWB ) / FW
        IF ( RMBH(1) .GE. 1.0 ) RMBH(1) = 0.0
      ENDIF
      IF ( DWW .EQ. 0.0 .OR. DWB .EQ. 0.0 ) THEN
        RMBH(2) = 0.0
      ELSE
        RMBH(2) = DWB / ( DWW + DWB )
      ENDIF
      DO 20, K = 1, 4
        DO 20, I = 1, N(K)
          FW = DISKS(I,3,K)
          DWW = DISKS(I,4,K)
          DWB = DISKS(I,5,K)
          IF ( FW .EQ. 0.0 .OR. DWW .EQ. 0.0
     X         .OR. DWB .EQ. 0.0 ) THEN
            R(I,1,K) = 0.0
          ELSE
            R(I,1,K) = ( DWW + DWB ) / FW
            IF ( R(I,1,K) .GE. 1.0 ) THEN
              IDB(1) = I
              IDB(2) = K
```

```
              RDB = R(I,1,K)
              R(I,1,K) = 0.0
            ENDIF
          ENDIF
          IF ( DWW .EQ. 0.0 .OR. DWB .EQ. 0.0 ) THEN
            R(I,2,K) = 0.0
          ELSE
            R(I,2,K) = DWB / ( DWW + DWB )
          ENDIF
20      CONTINUE
C
C       Estimate missing R's
C
        DO 35, J = 1, 2
          II = 0
          DO 30, K = 1, 4
            DO 30, I = 1, N(K)
              II = II + 1
              BUF(II,J) = R(I,J,K)
30        CONTINUE
35      CONTINUE
        DO 50, J = 1,2
          DO 50, I = 2, II - 1
            IF ( BUF(I,J) .EQ. 0.0 .AND.
     X           BUF(I-1,J) .NE. 0.0 .AND.
     X           BUF(I+1,J) .NE. 0.0 )
     X        BUF(I,J) = ( BUF(I-1,J) + BUF(I+1,J) ) / 2
50      CONTINUE
        DO 60, J = 1, 2
          II = 0
          DO 65, K = 1, 4
            DO 65, I = 1, N(K)
              II = II + 1
              IF ( R(I,J,K) .EQ. 0.0 )
     X          R(I,J,K) = BUF(II,J)
65        CONTINUE
60      CONTINUE
        RETURN
        END
```

Subroutines STEM and NSTEM

The computations for the dry weight of bark and wood are handled separately for stem components and non-stem components.

Stem Components

These include the merchantable stem, the non-merchantable stem ($\geq$ 2 cm dob), the stem top ($<$ 2 cm dob), and the stump.

Subroutine STEM is used for calculating the desired biomass information by using the volume-weighted ratios R_i and r_i for the merchantable stem, the nonmerchantable stem, and the top stem as defined by the user: The subroutine STEM also calculates the biomass information for the stump by using the simple unweighted ratios R_i and r_i.

The volume of merchantable and nonmerchantable stem sections is calculated by the Smalian formula:

$$V_i = \frac{\pi}{8} (DOB_i^2 + DOB_{i+1}^2)(H_{i+1} - H_i)$$

where (DOB_i, H_i) and (DOB_{i+1}, H_{i+1}) are the tree diameter outside bark and tree height at each end of the stem section. The summation $V = \Sigma V_i$ is the total volume of a specified tree component.

The volume-weighted ratios R_i and r_i are obtained as follows:

$$R_i = \Sigma \left[\left(\frac{R_i + R_{i+1}}{2} \right) V_i \right] / V$$

$$r_i = \Sigma \left[\left(\frac{r_i + r_{i+1}}{2} \right) V_i \right] / V$$

The estimate for the total dry weight (DW) of a tree component and its partition into dry weight of bark (DWB) and dry weight of wood (DWW) is computed from the total fresh weight (FW) as

$$DW = R_i \ (FW)$$

$$DWB = r_i \ (DW), \text{ and } DWW = DW - DWB$$

A listing of the STEM subroutine is given below.

```
      SUBROUTINE STEM(N,STUMP,MBH,DHT,TH,RSTUMP,
     X              RMBH,R,RAV,VOL,COMP,RC)
C
C     This subroutine calculates dry weights and volumes
C     of stem components of tree
C
C     Parameters
C
C     N       - number of disks
C               N(1,j) - 2 + number of sample disks in jth
C                        component
C               N(2,j) - number of sample disks used by subroutine
C                        ( returned )
C
C     STUMP       - top stump sample disk
C               STUMP(1) - diameter over bark of top stump disk
C               STUMP(2) - height
C
C     MBH         - mbh disk
C               MBH(1) - diameter over bark of mbh disk
C               MBH(2) - height
C
C     DHT         - heights and diameters of sample disks
C               DHT(i,1,k) - ith diameter over bark for kth component
C               DHT(i,2) - ith height for kth component
C
C     TH        - total height of tree
C
C     RSTUMP        - ratios for stump
C               RSTUMP(1) - ( dry-wt / fresh-wt ) for top stump
C                           sample disk
C               RSTUMP(2) - ( dry-bark / dry-wt )
C
C     RMBH        - ratios for mbh disk
C               RMBH(1) - ( dry-wt / fresh-wt )
C               RMBH(2) - ( dry-bark / dry-wt )
C
C     R       - ratios
C               R(i,1,k) - ith disk ( dry-wt / fresh-wt ) for
C                          kth component
C               R(i,2,k) - ith disk ( dry-bark / dry-wt ) for
C                          kth component
C
C     RAV         - average R
C               RAV(i,1,k) average of R(i,1,k) and R(i+1,1,k)
C               RAV(i,2,k) average of R(i,2,k) and R(i+1,2,k)
C
C     VOL       - volume
C               VOL(i,k) volume of section between ith and
C               i+1th disks on kth component
C
C     COMP        - weights and volumes of tree components
C               COMP(1,k) - fresh weight of kth tree component
C               COMP(2,k) - dry weight wood of kth tree component
C               COMP(3,k) - dry weight bark of kth tree component
C               COMP(4,k) - volume of kth tree component
C
C     RC        - return code
C               RC(k) - return code for kth component
C                     = 0 if dry weights calculated
C                     = 1 if component is missing
C                     = 2 if insufficient data for component
C
      INTEGER N(2,3)
      REAL STUMP(2), MBH(2), DHT(7,2,3), TH
      REAL RSTUMP(2), RMBH(2), R(7,2,3), RAV(6,2,3)
      REAL VOL(6,3), COMP(4,3)
      INTEGER RC(3)
C
      REAL VP, CH
C
C     Do merchantable stem
C
      DHT(1,1,1) = STUMP(1)       ! get top disk from stump
      DHT(1,2,1) = STUMP(2)
      R(1,1,1) = RSTUMP(1)
      R(1,2,1) = RSTUMP(2)
C
      DHT(2,1,1) = MBH(1)         ! get disk taken at MBH
      DHT(2,2,1) = MBH(2)
      R(2,1,1) = RMBH(1)
      R(2,2,1) = RMBH(2)
C
      IF ( DHT(1,1,1) .LE. 0.1 .AND.
     X     DHT(N(1,1),1,1) .EQ. 0.0 ) THEN
C
C       tree is a small tree
C
        DO 10, I = 1, 7
          R(I,1,1) = 0.0
          R(I,2,1) = 0.0
          IF ( I .GE. 7 ) GO TO 10
            RAV(I,1,1) = 0.0
            RAV(I,2,1) = 0.0
            VOL(I,1) = 0.0
10      CONTINUE
        COMP(2,1) = 0.0
        COMP(3,1) = 0.0
```

```
      COMP(4,1) = 0.0
      RC(1) = 1
    ELSE
      IF ( MBH(1) .LE. 0.1 ) THEN
C
C       MBH not in merchantable stem
C
        DHT(2,1,1) = 0.0
        DHT(2,2,1) = 0.0
      ENDIF
      CALL VDRYWB(N(1,1),DHT(1,1,1),R(1,1,1),
     X            RAV(1,1,1),VOL(1,1),COMP(1,1),RC(1))
    ENDIF
C
C   Do nonmerchantable stem
C
    IF ( N(2,1) .EQ. 0 ) N(2,1) = N(1,1)
    DHT(1,1,2) = DHT(N(2,1),1,1)        ! get highest disk in
    DHT(1,2,2) = DHT(N(2,1),2,1)        ! merchantable stem
    R(1,1,2) = R(N(2,1),1,1)
    R(1,2,2) = R(N(2,1),2,1)
    IF ( DHT(1,1,2) .EQ. 0.0 .OR.
   X     DHT(1,2,2) .EQ. 0.0 ) THEN
      DHT(1,1,2) = STUMP(1)         ! use stump if small
      DHT(1,2,2) = STUMP(2)         ! tree
      R(1,1,2) = RSTUMP(1)
      R(1,2,2) = RSTUMP(2)
    ENDIF
    DHT(2,1,2) = MBH(1)
    DHT(2,2,2) = MBH(2)
    R(2,1,2) = RMBH(1)
    R(2,2,2) = RMBH(2)
    IF ( MBH(1) .GT. 0.1 .OR. MBH(1) .LT. 0.02 ) THEN
C
C     MBH not in nonmerchantable stem
C
      DHT(2,1,2) = 0.0
      DHT(2,2,2) = 0.0
    ENDIF
    CALL VDRYWB(N(1,2),DHT(1,1,2),R(1,1,2),
   X            RAV(1,1,2),VOL(1,2),COMP(1,2),RC(2))
C
C   Do nonmerchantable top
C
    DHT(1,1,3) = DHT(N(2,2),1,2)        ! get highest disk in
    DHT(1,2,3) = DHT(N(2,2),2,2)        ! nonmerchantable stem
    R(1,1,3) = R(N(2,2),1,2)
    R(1,2,3) = R(N(2,2),2,2)
    DHT(2,1,3) = 0.0
    DHT(2,2,3) = 0.0
    CALL VDRYWB(N(1,3),DHT(1,1,3),R(1,1,3),
   X            RAV(1,1,3),VOL(1,3),COMP(1,3),RC(3))
    IF ( RC(3) .EQ. 0 ) THEN
      CH = TH - DHT(N(2,3),2,3)
      IF ( CH .GT. 0 ) THEN
        VP = 0.392699 * CH * DHT(N(2,3),1,3) ** 2
        COMP(4,3) = COMP(4,3) + VP
      ENDIF
    ENDIF
    RETURN
    END
    SUBROUTINE VDRYWB(N,DHT,R,RAV,VOL,COMP,RC)
C
    INTEGER N(2)
    REAL DHT(7,2), R(7,2), RAV(6,2), VOL(6), COMP(4)
    INTEGER RC
C
    INTEGER I, J
    REAL TEMP, RV(2)
    LOGICAL FLAG
C
C   Initialize
C
    N(1) = MOD(N(1),7)
    IF ( N(1) .EQ. 0 ) N(1) = 7
    N(2) = N(1)
    DO 10, I = 1, 6
      RAV(I,1) = 0.0
      RAV(I,2) = 0.0
      VOL(I) = 0.0
10  CONTINUE
    DO 20, I = N(1), 7
      IF ( I .EQ. N(1) ) GOTO 20
      DHT(I,1) = 0.0
      DHT(I,2) = 0.0
      R(I,1) = 0.0
      R(I,2) = 0.0
20  CONTINUE
    IF ( DHT(1,1).NE.0.0 .AND. DHT(N(1),1).NE.0.0 .AND.
   X     DHT(1,2).NE.0.0 .AND. DHT(N(1),2).NE.0.0 .AND.
   X          N(1).GE.2) THEN
C
C     Sufficient data
C
      RC = 0
C
C     Sort disks by heights, throw out bad disks
C
      DO 30, I = 1, N(1) - 1
        DO 30, J = 1, N(1) - 1
          IF ( (DHT(J,2) .GT. DHT(J+1,2) .AND.
   X            DHT(J+1,2) .NE. 0.0 )  .OR.
   X           DHT(J,1) .EQ. 0.0 .OR.
   X           DHT(J,2) .EQ. 0.0 ) THEN
            DO 25, K = 1, 2
              TEMP = DHT(J,K)
              DHT(J,K) = DHT(J+1,K)
              DHT(J+1,K) = TEMP

              TEMP = R(J,K)
              R(J,K) = R(J+1,K)
              R(J+1,K) = TEMP
25          CONTINUE
          ENDIF
30      CONTINUE
      DO 40, I = 1, N(1)
        IF ( DHT(1,1) .EQ. 0.0 .OR. DHT(I,2) .EQ. 0.0 ) THEN
          N(2) = N(2) - 1
        ENDIF
40    CONTINUE
      DO 50, I = N(2), N(1)
        IF ( I .EQ. N(2) ) GOTO 50
        DHT(I,1) = 0.0
        DHT(I,2) = 0.0
        R(I,1) = 0.0
        R(I,2) = 0.0
50    CONTINUE
      IF ( N(2) .GE. 2) THEN
C
C       Enough good disks, estimate any missing R's
C
        DO 60, I = 2, N(2)
          DO 60, J = 1, 2
            IF ( R(I,J) .EQ. 0.0 ) R(I,J) = R(I,J)
            IF ( R(N(2),J) .EQ. 0.0 )
   X                R(N(2),J) = R(N(2)+1-I,J)
60      CONTINUE
        DO 70, I = 2, N(2) - 1
          DO 70, J = 1, 2
            IF ( R(I,J) .EQ. 0.0 ) THEN
              R(I,J) = R(I-1,J) + R(I+1,J)
              IF ( R(I-1,J) .NE. 0.0 .AND.
   X               R(I+1,J) .NE. 0.0 ) THEN
                R(I,J) = R(I,J) / 2
              ENDIF
            ENDIF
70      CONTINUE
        FLAG = .TRUE.
        DO 80, I = 1, N(2)
          IF (R(I,1).EQ. 0.0 ) FLAG = .FALSE.
          IF (R(I,2).EQ. 0.0 ) FLAG = .FALSE.
80      CONTINUE
        IF ( FLAG ) THEN
C
C         Enough R data
C
C         Calculate VOL, RAV for pairs of disks
C         Calculate total volume of part and
C           volume-weighted RV ratios for entire part
C
          COMP(4) = 0.0
          RV(1) = 0.0
          RV(2) = 0.0
          DO 90, I = 1, N(2) - 1
            RAV(I,1) = ( R(I,1) + R(I+1,1) ) / 2
            RAV(I,2) = ( R(I,2) + R(I+1,2) ) / 2
            VOL(I) = 0.392699 *
   X                 ( DHT(I+1,1)**2 + DHT(I,1)**2 )
   X               * ( DHT(I+1,2) - DHT(I,2) )
            COMP(4) = COMP(4) + VOL(I)
            RV(1) = RV(1) + RAV(I,1) * VOL(I)
            RV(2) = RV(2) + RAV(I,2) * VOL(I)
90        CONTINUE
          RV(1) = RV(1) / COMP(4)
          RV(2) = RV(2) / COMP(4)
          COMP(3) = COMP(1) * RV(1) * RV(2)
          COMP(2) = COMP(1) * RV(1) - COMP(3)
        ELSE
C
C         Insufficient data
C
          COMP(2) = 0.0
          COMP(3) = 0.0
          COMP(4) = 0.0
          RC = 2
        ENDIF
      ELSE
C
C         Insufficient data
C
        COMP(2) = 0.0
        COMP(3) = 0.0
        COMP(4) = 0.0
        RC = 2
      ENDIF
    ELSE
C
C     Insufficient data
C
      COMP(2) = 0.0
      COMP(3) = 0.0
      COMP(4) = 0.0
      RC = 2
    ENDIF
    RETURN
    END
```

Nonstem Components

These include the large live branches, the small live branches including foliage, the dead branches, and the cones.

Subroutine NSTEM computes dry weights of bark and wood for live branches, dry weight of wood for dead branches, and dry weight of cones. Live branches are grouped into branches $\geq$ 2 cm dob and branches $<$ 2 cm dob. Size limits are arbitrary and can be adjusted for any other specification. Small branches can be further split into groups of dry weight of bark and wood and groups of dry weight of foliage.

Computations are similar to those described under the STEM subroutine; the main difference is that no weighting is done for nonstem components and only the simple ratio averages of R_i and r_i are used in computing biomass weights.

A listing of this subroutine is given below.

```
      SUBROUTINE NSTEM(N,SW,R,PART)
C
C     This subroutine calculates dry weights
C     of nonstem components of the tree. There
C     is no volume estimation for these
C     components.
C
C     Nonstem components:
C
C        Large live branches
C        Small live branches + foliage
C        Dead branches
C        Cones
C
C           For large and small live branches
C     the dry weight of wood and dry weight of bark
C     or needles are calculated.  For dead branches
C     and cones only total dry weight is calculated.
C
C     Parameters
C
C     N       - number of samples
C               N(1) - number of samples for stump
C                      ( 1st tree component )
C               N(2) - number of samples for large
C                      branches ( 2nd tree component )
C               N(3) - . . . . small branches ( 3rd tree component )
C               N(4) - . . . . dead branches ( 4th tree component )
C               N(5) - . . . . cones ( 5th tree component )
C
C     SW      - sample weights
C               SW(i,1,k) - ith sample fresh-wt for kth
C                  tree component
C               SW(i,2,k) - ith sample dry-wt wood or
C                  sample dry-wt for kth tree component
C               SW(i,3,k) - ith sample dry-wt bark for
C                  kth tree component ( not always used )
C
C     R       - ratios
C               R(i,1,k) - ith sample ( dry-wt / fresh-wt )
C                  for kth tree component
C               R(i,2,k) - ith sample ( dry-bark / dry-wt )
C                  for kth tree component
C
C     PART      - weights of tree components
C               PART(1,k) fresh weight of kth tree component
C               PART(2,k) dry weight wood or dry weight
C                  of kth tree component
C               PART(3,k) dry weight bark of kth tree
C                  component or unused
C
C     Subroutines called:
C
C           DRYWB
C           DRYW
C
      INTEGER N(5)
      REAL SW(5,3,5), R(5,2,5), PART(3,5)
C
C     Do stump
C
      CALL DRYWB(N(1),SW(1,1,1),R(1,1,1),PART(1,1))
C
C     Do large branches
C
      CALL DRYWB(N(2),SW(1,1,2),R(1,1,2),PART(1,2))
C
C     Do small branches
C
      CALL DRYWB(N(3),SW(1,1,3),R(1,1,3),PART(1,3))
C
C     Do dead branches
C
      CALL DRYW(N(4),SW(1,1,4),R(1,1,4),PART(1,4))
C
C     Do cones
C
      CALL DRYW(N(5),SW(1,1,5),R(1,1,5),PART(1,5))
      RETURN
      END
      SUBROUTINE DRYWB(N,SW,R,PART)
C
      INTEGER N
      REAL SW(5,3), R(5,2), PART(3)
C
      INTEGER I, J, NR
      REAL RAV(2)
C
C     Calculate R
C
      NR = N
      DO 20, I = 1, N
        IF ( SW(I,1) .EQ. 0.0 .OR. SW(I,2) .EQ. 0.0
     X       .OR. SW(I,3) .EQ. 0.0 ) THEN
          R(I,1) = 0.0
          R(I,2) = 0.0
          NR = NR -1
        ELSE
          R(I,1) = ( SW(I,2) + SW(I,3) ) / SW(I,1)
          R(I,2) = SW(I,3) / ( SW(I,2) + SW(I,3) )
          IF ( R(I,1) .GE. 1.0 ) THEN
            R(I,1) = 0.0
            R(I,2) = 0.0
            NR = NR - 1
          ENDIF
        ENDIF
   20 CONTINUE
C
C     Calculate average R
C
      RAV(1) = 0.0
      RAV(2) = 0.0
      IF ( NR .NE. 0 ) THEN
        DO 40, J = 1, 2
          DO 40, I = 1, N
            RAV(J) = RAV(J) + R(I,J)
   40   CONTINUE
        RAV(1) = RAV(1) / NR
        RAV(2) = RAV(2) / NR
      ENDIF
C
C     Calculate dry weights
C
      PART(3) = PART(1) * RAV(1) * RAV(2)
      PART(2) = PART(1) * RAV(1) - PART(3)
      RETURN
      END
      SUBROUTINE DRYW(N,SW,R,PART)
C
      INTEGER N
      REAL SW(5,2), R(5), PART(2)
C
      INTEGER I, J, NR
      REAL RAV
C
C     Calculate R
C
      NR = N
      DO 20, I = 1, N
        IF ( SW(I,1) .EQ. 0.0 .OR.
     X       SW(I,2) .EQ. 0.0 ) THEN
          R(I) = 0.0
          NR = NR -1
        ELSE
          R(I) = SW(I,2) / SW(I,1)
          IF ( R(I) .GE. 1.0 ) THEN
            R(I) = 0.0
            NR = NR - 1
          ENDIF
        ENDIF
   20 CONTINUE
C
C     Calculate average R
C
      RAV = 0.0
      IF ( NR .NE. 0 ) THEN
        DO 40, I = 1, N
          RAV = RAV + R(I)
   40   CONTINUE
        RAV = RAV / NR
      ENDIF
C
C     Calculate dry weights
C
      PART(2) = PART(1) * RAV
      RETURN
      END
```

REFERENCES

Alemdag, I.S. 1980. Manual of data collection and processing for the development of forest biomass relationships. Environ. Can., Can. For. Serv. Petawawa Nat. For. Inst. Chalk River, Ontario. Inf. Rep. PI-X-4.

Singh, T. 1982. Biomass equations for ten major tree species of the prairie provinces. Environ. Can., Can. For. Serv., North. For. Res. Cent. Edmonton, Alberta. Inf. Rep. NOR-X-242.

ACKNOWLEDGMENTS

The assistance of N. Leenders, C. Martin, and S. Lux in the formulation, writing, and testing of the preliminary programs is gratefully acknowledged.

T. Singh
D. Campbell
November 1983

APPENDIX 1

Listed below are sample programs used for driving the subroutines RATIOS, STEM, and NSTEM at NoFRC.

```
C
C     This program drives the subroutine RATIOS
C
      INTEGER TREEN, N(4), IDB(2), I, J, K, II
      REAL MBH(5), DISCS(5,5,4), RMBH(2), R(5,2,4), RDB
      DATA N/ 1,4,3,1/
      OPEN (UNIT=1,FILE='BIOB1.DAT',RECORDSIZE=410,STATUS='OLD')
      OPEN (UNIT=2,FILE='BIGRB1.TEJ',CARRIAGECONTROL='LIST',
     X            STATUS='NEW')
      OPEN (UNIT=3,FILE='RGB1.TMP',CARRIAGECONTROL='LIST',
     X            STATUS='NEW')
      OPEN (UNIT=4,FILE='SMALLRB1.TEJ',CARRIAGECONTROL='LIST',
     X            STATUS='NEW')
1     CONTINUE
      READ (1,400,END=999) TREEN, ( DISCS(1,J,1), J=1,5 ),
     X ( MBH(J), J=1,5 ), ( ( DISCS(I,J,2), J=1,5 ), I=1,4 ),
     X ( ( DISCS(I,J,3), J=1,5 ), I=1,3 ),
     X ( DISCS(1,J,4), J=1,5 )
      CALL RATIOS(N,MBH,DISCS,RMBH,R,IDB,RDB)
      IF ( IDB(1) .NE. 0 ) THEN
        II = 0
        DO 30, K = 1, IDB(2)
          II = II + N(K)
30      CONTINUE
        II = II +IDB(1) - N(IDB(2))
        IF ( IDB(2) .GT. 1 ) II = II + 1
        WRITE (3,127) TREEN, DISCS(IDB(1),4,IDB(2)),
     X    DISCS(IDB(1),5,IDB(2)), DISCS(IDB(1),3,IDB(2)),
     X    II, RDB
      ENDIF
      WRITE (2,410) TREEN,R(1,1,1),RMBH(1),
     X ( R(I,1,2), I = 1, 4 ),
     X ( R(I,1,3), I = 1, 3 ), R(1,1,4)
      WRITE (4,410) TREEN,R(1,2,1),RMBH(2),
     X ( R(I,2,2), I = 1, 4 ),
     X ( R(I,2,3), I = 1, 3 ), R(1,2,4)
      GOTO 1
999   CONTINUE
127   FORMAT (1X,I3,3(2X,F5.1),2X,I3,2X,F7.4)
400   FORMAT (I3,T77,4(F4.3,F4.2,3F5.1),2(F3.3,F4.2,3F5.1),
     X (F3.3,F4.2,2F5.1,F4.1),3(F3.3,F4.2,3F4.1))
410   FORMAT (I3,10(2X,F7.6))
      STOP
      END


C
C     This program drives the subroutine STEM
C
      INTEGER TREEN, N(2,3)
      REAL TH, STUMP(2), MBH(2), DHT(7,2,3)
      REAL RSTUMP(2), RMBH(2), R(7,2,3)
      REAL RAV(6,2,3), VOL(6,3), COMP(4,3)
      INTEGER RC(3)
C
      INTEGER I, J, K
C
C     Initialize
C
      DATA N/6,0,5,0,3,0/
C
C     Open files
C
      OPEN (UNIT=1,FILE='SMALLRB1.TEJ',STATUS='OLD')
      OPEN (UNIT=2,FILE='BIGRB1.TEJ',STATUS='OLD')
      OPEN (UNIT=3,FILE='BIOB1.DAT',RECORDSIZE=410,
     X            STATUS='OLD')
      OPEN (UNIT=4,FILE='TMP1.TEJ',CARRIAGECONTROL='LIST',
     X            STATUS='NEW')
      OPEN (UNIT=7,FILE='RESIB1.TEJ',CARRIAGECONTROL='LIST',
     X            STATUS='NEW')
      OPEN (UNIT=8,FILE='TMP2II.TEJ',CARRIAGECONTROL='LIST',
     X            STATUS='NEW')
      OPEN (UNIT=9,FILE='TMPII.TEJ',CARRIAGECONTROL='LIST',
     X            STATUS='NEW')
      OPEN (UNIT=10,FILE='RESIIB1.TEJ',CARRIAGECONTROL='LIST',
     X            STATUS='NEW')
      OPEN (UNIT=11,FILE='TMPIII.TEJ',CARRIAGECONTROL='LIST',
     X            STATUS='NEW')
      OPEN (UNIT=12,FILE='RESIIIB1.TEJ',CARRIAGECONTROL='LIST',
     X            STATUS='NEW')
C
C     Read and process data until end of file
C
1     CONTINUE
      READ (3,400,END=999) TREEN, TH, ( COMP(1,K), K = 1, 3 ),
     X            STUMP, MBH, ( ( DHT(I,J,1), J = 1, 2 ), I = 3, 6
     X            ( ( DHT(I,J,2), J = 1, 2 ), I = 3, 5 ),
     X            ( DHT(3,J,3), J = 1, 2 )
      READ (2,410) RSTUMP(1),RMBH(1),( R(1,1,1), I=3,6 ),
     X            ( R(I,1,2), I=3,5 ), R(3,1,3)
      READ (1,410) RSTUMP(2),RMBH(2),( R(1,2,1), I=3,6 ),
     X            ( R(I,2,2), I=3,5 ), R(3,2,3)
      CALL STEM(N,STUMP,MBH,DHT,TH,RSTUMP,RMBH,R,RAV,
     X            VOL,COMP,RC)
      IF ( RC(1) .EQ. 0 ) THEN
        WRITE (4,430) TREEN,(RAV(I,2,1),RAV(I,1,1),VOL(I,1), I=1,5)
        WRITE (7,440) TREEN,(R(1,2,1),I=1,6),(R(I,1,1),I=1,6),
     X            COMP(4,1), COMP(3,1), COMP(2,1)
      ELSE
        IF ( RC(1) .EQ. 1 ) THEN
          WRITE(4,431) TREEN
          WRITE(7,441) TREEN,0.0,0.0,0.0,0.0
        ELSE
          WRITE(4,432) TREEN
          WRITE(7,432) TREEN
        ENDIF
      ENDIF
      IF ( RC(2) .EQ. 0 ) THEN
        WRITE (9,450) TREEN,(RAV(I,2,2),RAV(I,1,2),VOL(I,2), I=1,4)
        WRITE (10,460) TREEN,(R(1,2,2),I=1,5),(R(I,1,2),I=1,5),
     X            COMP(4,2), COMP(3,2), COMP(2,2)
        WRITE (8,465) TREEN, ( ( DHT(I,J,2), J = 1, 2 ), I = 1,5 )
      ELSE
        WRITE (9,432) TREEN
        WRITE (10,432) TREEN
        WRITE (8,432) TREEN
      ENDIF
      IF ( RC(3) .EQ. 0 ) THEN
        WRITE (11,470) TREEN,RAV(1,2,3),RAV(1,1,3)
        WRITE (12,480) TREEN, COMP(4,3), COMP(3,3), COMP(2,3)
      ELSE
        WRITE (11,432) TREEN
        WRITE (12,432) TREEN
      ENDIF
      GOTO 1
999   CONTINUE
400   FORMAT ( I3,6X,F4.2,T42,F6.2,F5.2,F4.2,
     X            T77,4(F4.3,F4.2,15X),2(F3.3,F4.2,15X),
     X            F3.3,F4.2,14X,3(F3.3,F4.2,12X) )
410   FORMAT (3X,10(2X,F7.6))
430   FORMAT (1X,I3,5(2X,F6.5),F8.4))
431   FORMAT (2X,I3,' is a small tree.')
432   FORMAT (2X,I3,' has insufficient data.')
440   FORMAT(1X,I3,12(1X,F6.5),3F11.5)
441   FORMAT (1X,I3,84X,3F11.5)
450   FORMAT (1X,I3,4(2X,F6.5),F8.4))
460   FORMAT(1X,I3,10(1X,F6.5),3F11.5)
465   FORMAT (1X,I3,10F12.4)
470   FORMAT (1X,I3,2(2X,F6.5))
480   FORMAT(1X,I3,3F11.5)
      STOP
      END


C
C     This program drives the subroutine NSTEM
C
      INTEGER TREEN, N(5)
      REAL SW(5,3,5), R(5,2,5), PART(3,5)
      INTEGER I, J, K
      DATA N/1,3,3,3,1/
      OPEN (UNIT=1,FILE='BIOB1.DAT',RECORDSIZE=410,
     X            STATUS='OLD')
      OPEN (UNIT=2,FILE='RESIVB1.TEJ',CARRIAGECONTROL='LIST',
     X            STATUS='NEW')
      OPEN (UNIT=3,FILE='RESVB1.TEJ',CARRIAGECONTROL='LIST',
     X            STATUS='NEW')
      OPEN (UNIT=4,FILE='RESVIB1.TEJ',CARRIAGECONTROL='LIST',
     X            STATUS='NEW')
      OPEN (UNIT=7,FILE='RESVIIB1.TEJ',CARRIAGECONTROL='LIST',
     X            STATUS='NEW')
      OPEN (UNIT=8,FILE='RESSDWB1.TEJ',CARRIAGECONTROL='LIST',
     X            STATUS='NEW')
1     CONTINUE
      READ (1,400,END=999) TREEN, ( PART(1,K), K = 1, 5),
     X  ( SW(1,J,1), J = 1, 3 ),
     X  ( ( ( SW(I,J,K), J = 1, 3 ), I = 1, 3 ), K = 2, 3 ),
     X  ( ( SW(I,J,4), J = 1, 2 ), I = 1, 3 ),
     X  ( SW(1,J,5), J = 1, 2 )
      CALL NSTEM(N,SW,R,PART)
      WRITE (2,600) TREEN,( ( R(I,J,2), I = 1, 3 ), J = 1, 2 ),
     X            PART(3,2), PART(2,2)
      WRITE (3,600) TREEN,( ( R(I,J,3), I = 1, 3 ), J = 1, 2 ),
     X            PART(3,3), PART(2,3)
      WRITE (4,610) TREEN, ( R(1,1,4), I = 1, 3 ), PART(2,4)
      WRITE (7,620) TREEN, PART(2,5)
      WRITE (8,620) TREEN, PART(3,1), PART(2,1)
      GOTO 1
999   CONTINUE
400   FORMAT(I3,T37,F5.2,T57,4F5.2,T85,3F5.1,
     X            T91,100X,100X,3(3F4.1),3(F5.1,2F4.1),
     X            T66,100X,100X,100X,3(2F5.1),2F5.1)
600   FORMAT (1X,I3,6(2X,F6.5),2F11.5)
610   FORMAT (1X,I3,3(2X,F7.5),F11.5)
620   FORMAT (1X,I3,2F11.5)
      STOP
      END
```

Canada

Northern Forest Research Centre
5320 - 122 Street
Edmonton, Alberta
T6H 3S5 (403) 435-7210

© Minister of Supply and Services Canada 1983

Cat. No. Fo29-2/22-1983E
ISBN 0-662-12762-5
ISSN 0714-1181

Environment Canada
Canadian Forestry Service

Environnement Canada
Service canadien des forêts

POOR

QUALITY

ORIGINAL